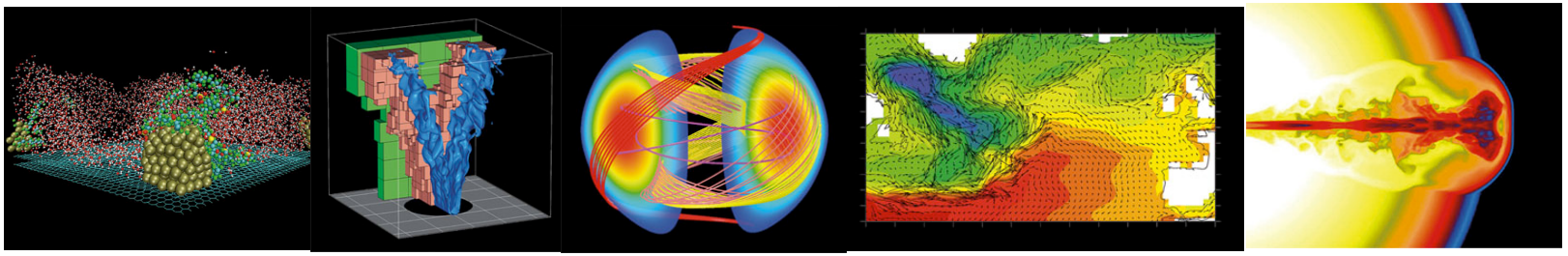


IO Performance Monitoring on Franklin

Katie Antypas
NERSC User Services Group
Kantypas@lbl.gov

Cray Technical Workshop
San Francisco, February 27, 2008

BERKELEY LAB

NERSC

Office of Science
U.S. DEPARTMENT OF ENERGY

NERSC User Services Group

- Help users debug and run jobs
- Help users optimize and scale codes
- Monitor system performance
- User education and training
- Research new software tools
- Participate in NERSC system procurements

Outline

- Why do IO Benchmarking?
- IO Performance Monitoring Franklin (XT4)
  - Dedicated vs Production Mode
  - Production Mode Variation
  - Performance Changes after System Upgrades

Challenges to IO Benchmarking

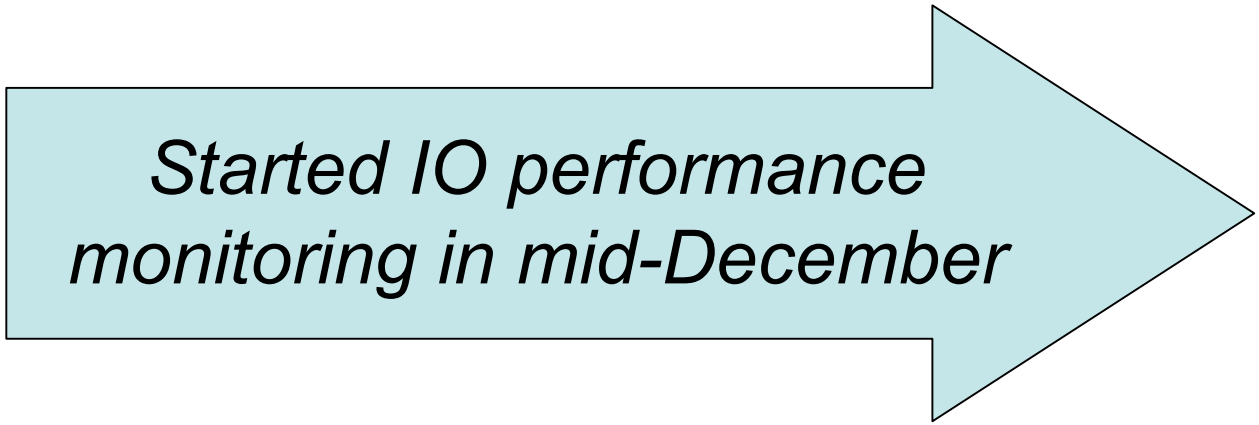
- **Results may be irrelevant within a few months, weeks or days**
  - **Systems changing very rapidly**
  - **Difficult to relate system changes back to benchmarking results**
- **Few will be able to reproduce results**
  - **Application parameters**
    - **Varying memory per node**
    - **Compiler differences**
  - **System software parameters**
    - **Different environment settings**
    - **Versions of software**
  - **Hardware configuration**
    - **Different ratios of compute nodes to IO nodes**

Reasons to Do IO Benchmarking

- From a research perspective
  - IO systems still not well understood
  - No model for interaction between applications, system software, hardware
- From NERSC users perspective
  - Not just a research exercise, IO performance crucial to science simulations
  - Better IO performance leaves more computational time for science
  - Applications have different IO access patterns, file sizes, parallel IO library interfaces
  - Users can make some adjustments but likely won't make wholesale changes to IO strategies.
  - Even if benchmarks are outdated and results can't be repeated, look for general patterns to help make recommendations for users
  - Feed results back for improvements in system configuration

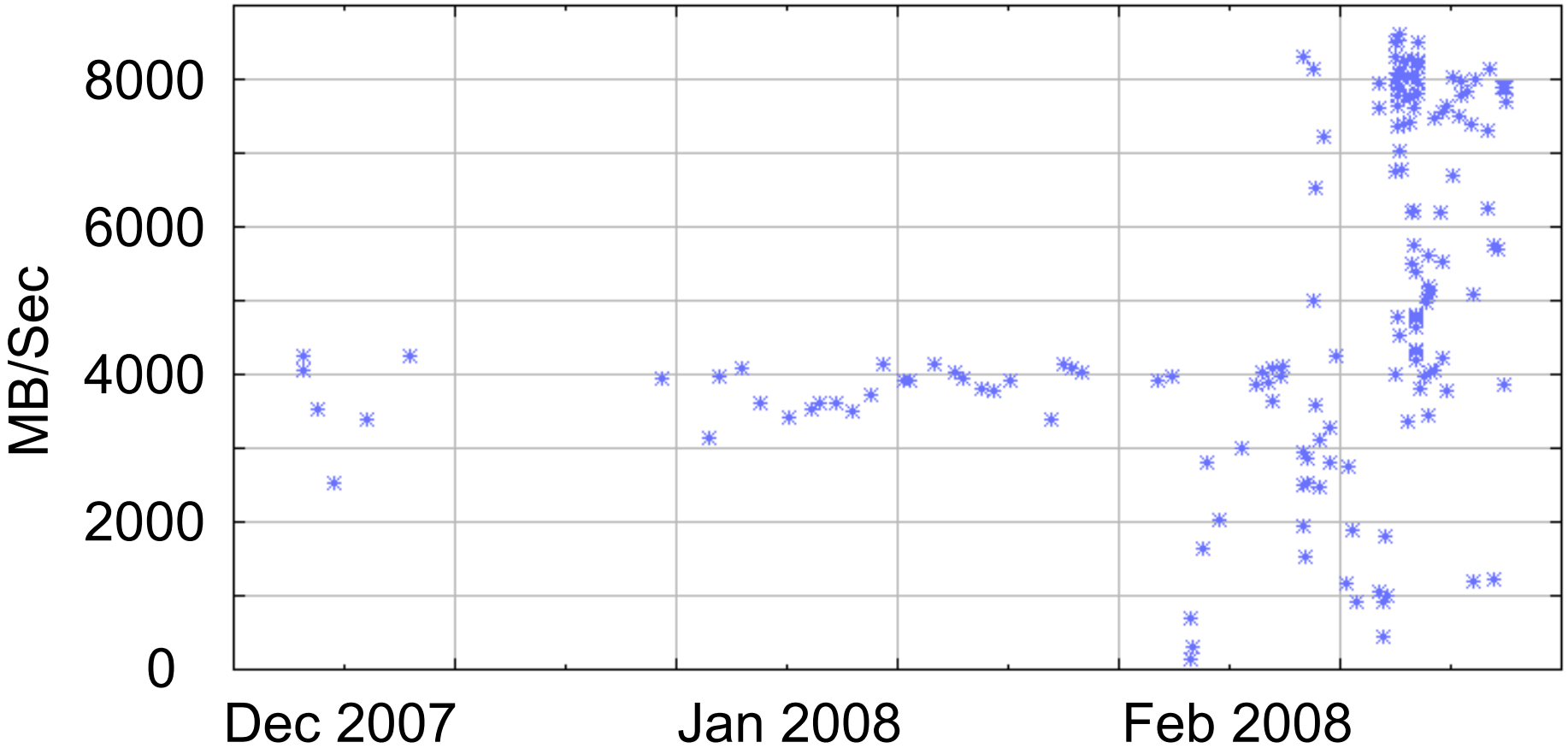
IO Performance Baseline

- Control case for IO benchmarking
- Comparison against system changes
- Systems are complicated. Not clear when one part of system changes how it will effect another
- Lustre is a shared resource. Performance depends on other jobs on the system



IO Performance Monitoring

64 Processor file-per-proc Write Test



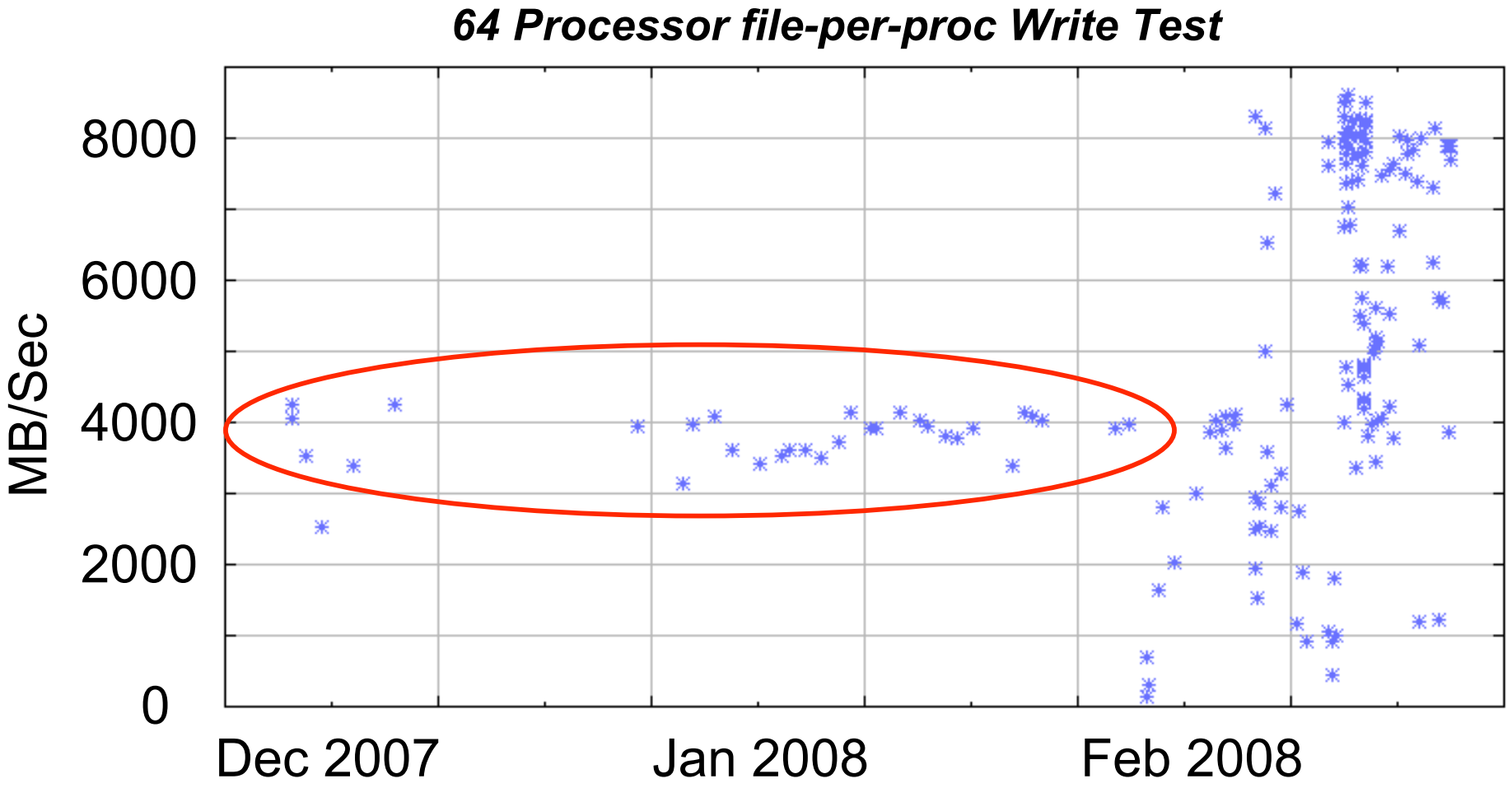
IOR Benchmark

- Developed at LLNL
- Highly parameterized believe it can mimic a number of full applications
  - Can do one file-per-processor or shared file IO
  - Multiple interfaces Posix, MPI-IO, HDF5 or Parallel-NetCDF interfaces
- Used in other NERSC procurements

IO Performance Monitoring

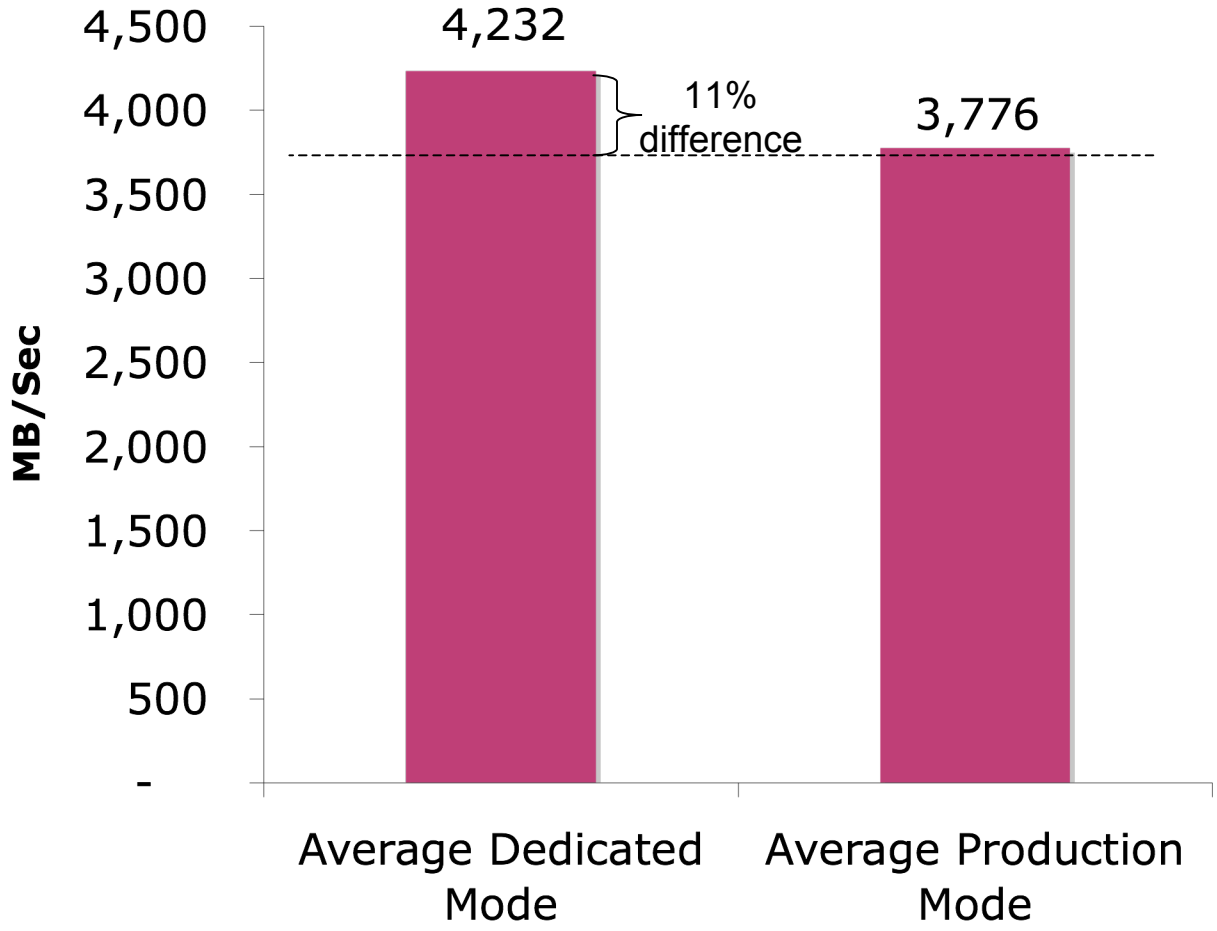
- Franklin is a production system and monitoring should not interfere with scientists' work
- Should be smallest possible run which still gives performance indication of the system
- Run same exact test every day
- Chosen Run
  - 64 processors, one file per proc, write performance
  - Each processor writes 1GB
  - Outside block buffer caching
  - Posix IO Interface
  - No Striping (Each file goes to own OST
    - lfs setstripe 0 1 -1

Performance Monitoring



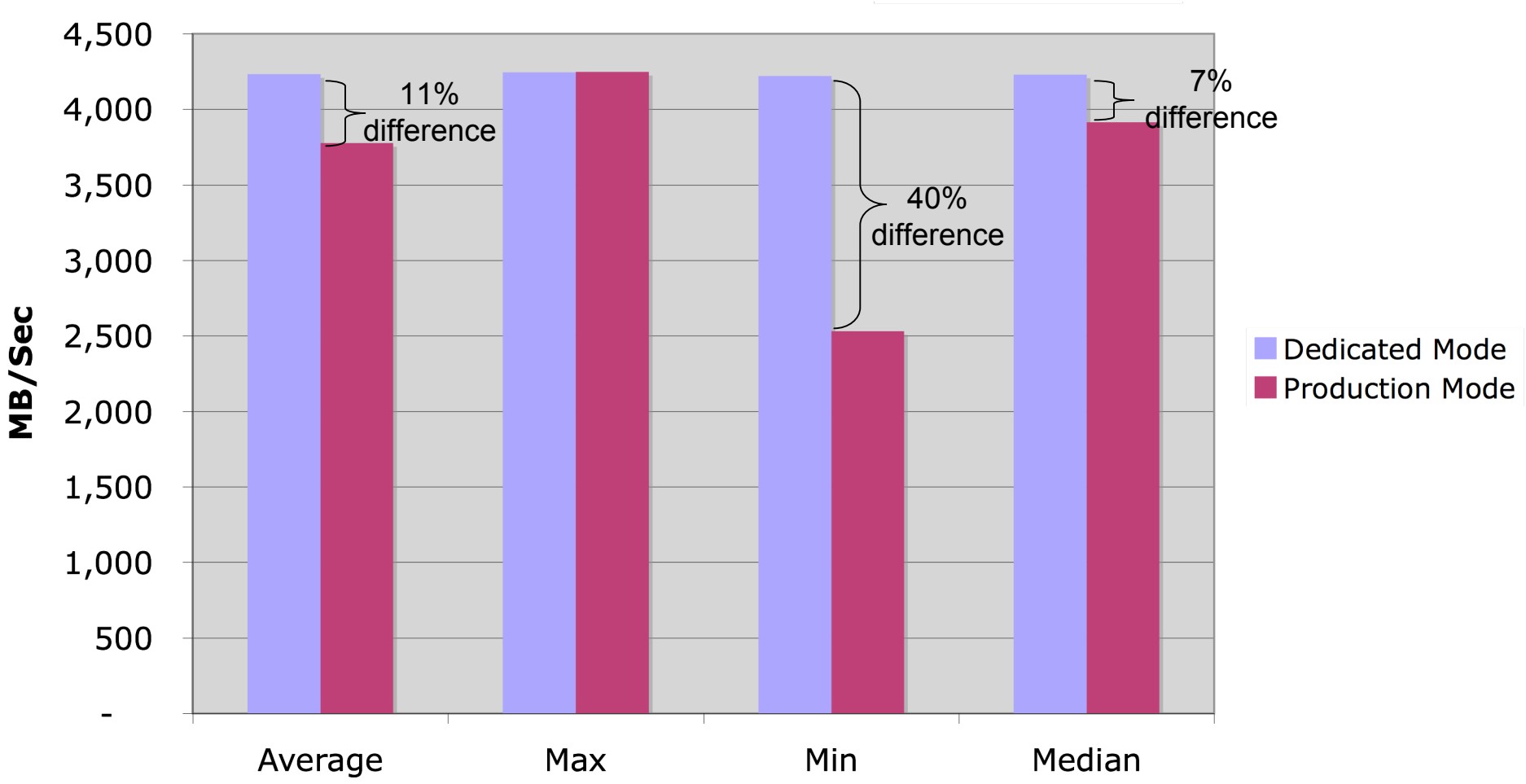
Relatively steady area from mid-December to early February had COV ~9.5%

Dedicated vs Production Mode

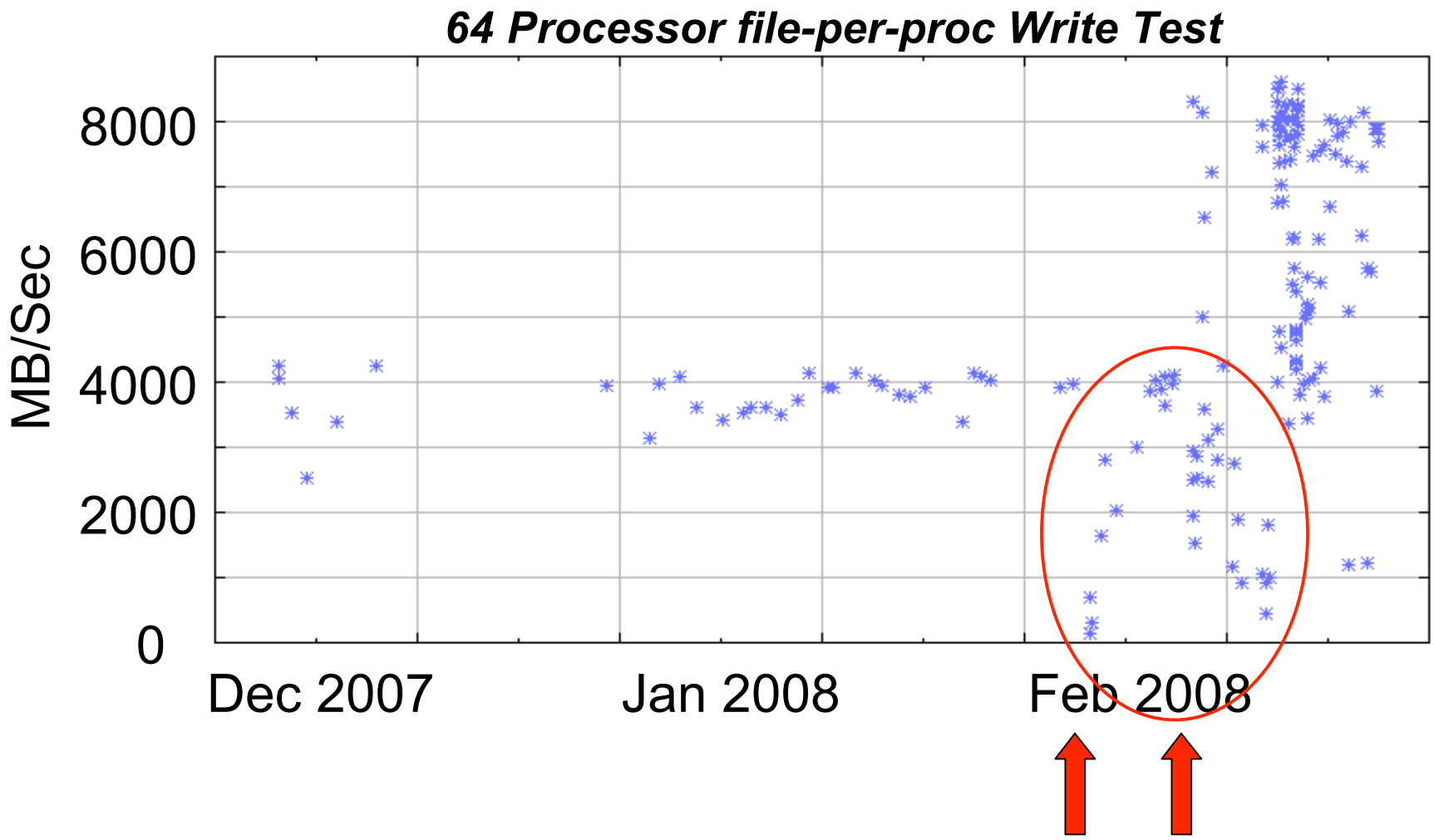


- *64 processor runs, file-per-processor, 1GB file per processor, write test*
- *Dedicated mode - only job on system*
- *Production mode - running with system full of jobs*
- *Production runs average of 31 runs over 5 weeks*
- *3 dedicated runs with less than 0.5% variation*
- *Performance lower by 11% during production mode*

Dedicated vs Production Mode

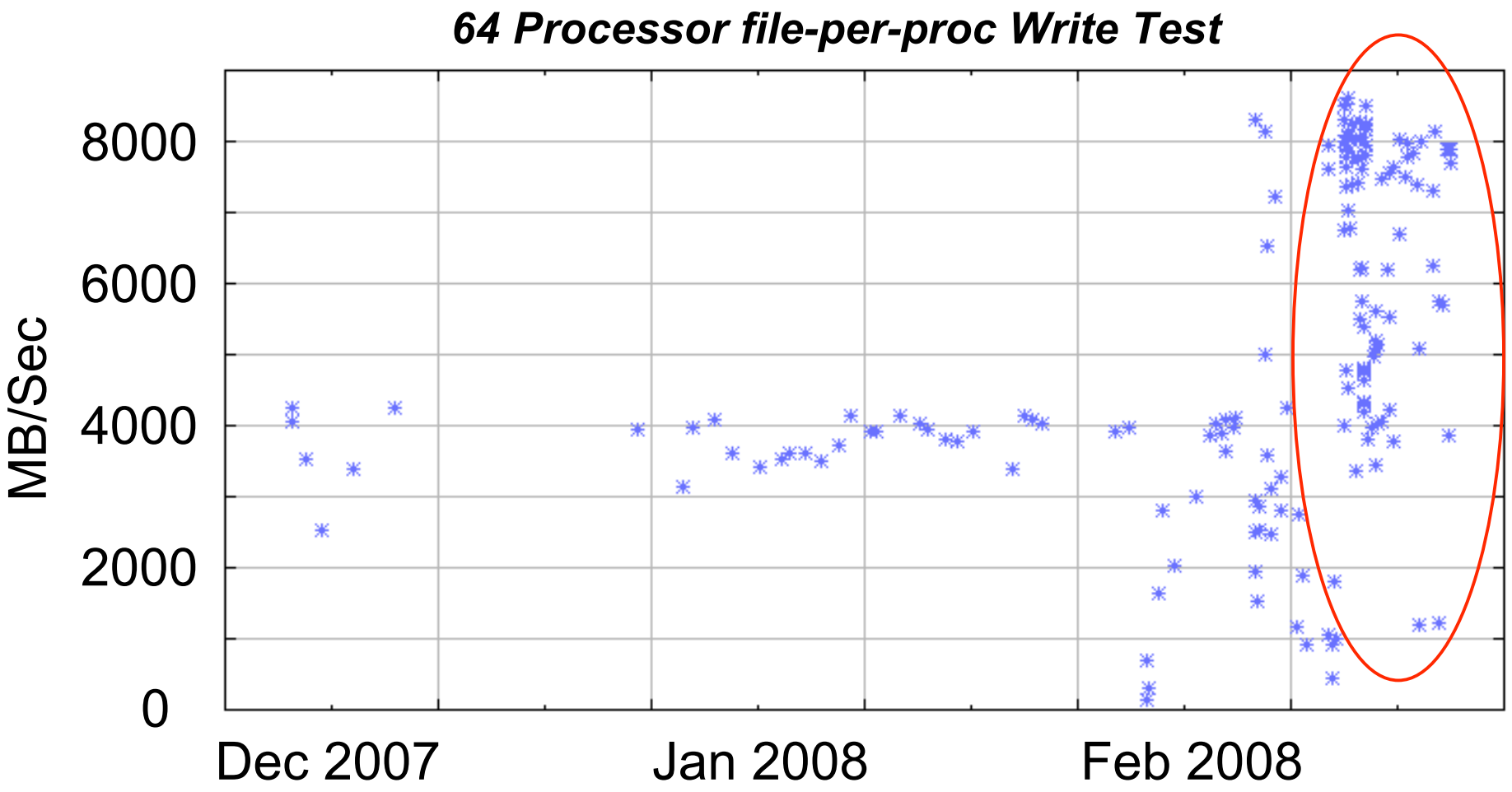


Temporary Decreased Performance After Upgrades



System upgrades on Feb 6th and Feb 13th

Increased Performance but Increased Variability

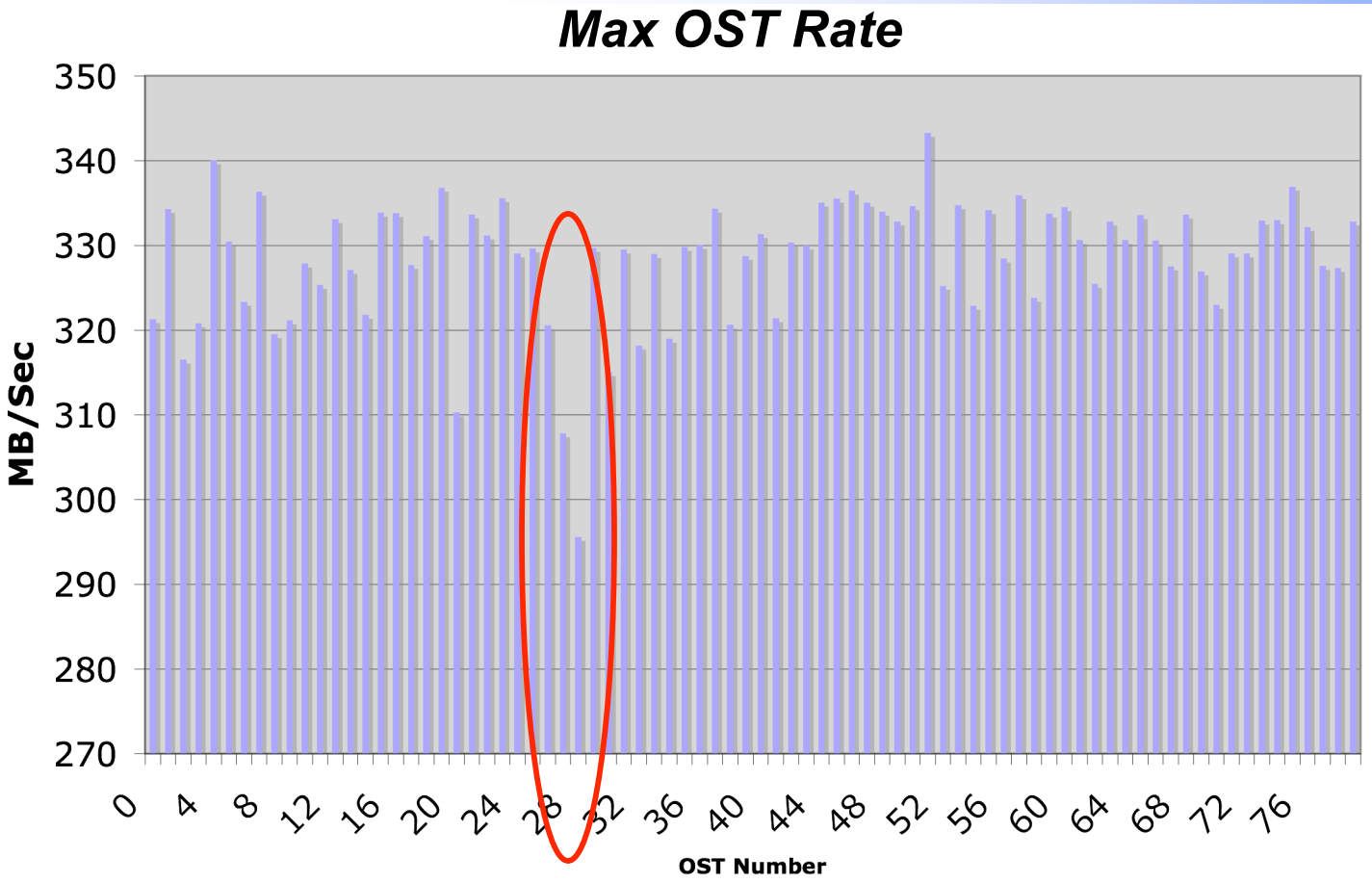


In the past week see a COV of 33%

What is cause of Variation?

- Recompile executable
- One or a few low performing OSTs?
- Interference from particularly IO intensive user?
- Verify no caching effect?
- Or did system upgrade increase performance and also variability?

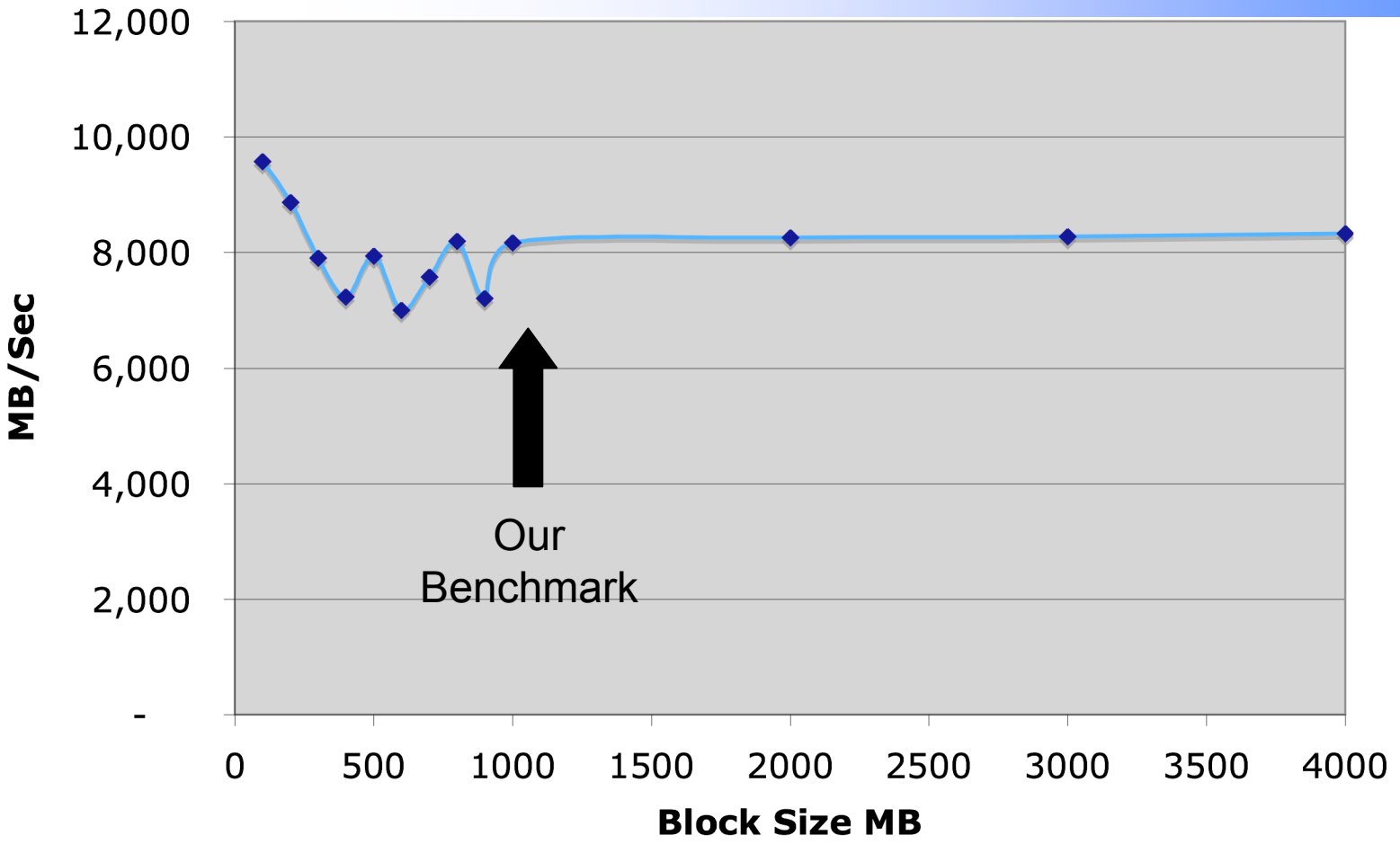
Check Individual OSTs



- *Write out 4 GB on a single processor to a single OST*
- *Set striping to start at specific OST (not round robin)*
- *9 runs, take maximum of all runs*

Although we may want to investigate the performance of OST 27 and 28, the difference can not explain the variation seen in IO benchmark

Verify No Caching Effect?



Each processor in benchmark is writing enough data to be above caching level. Caching not contributing to high level of variability

Conclusions

- Don't understand what is happening after upgrades
- Not certain reasons for recent increases in variation
- Before a full system upgrade, testing on smaller system helps, but often can not predict results of full system
- Helps expose issues we were not aware of
- Even if cause is unknown, identification is the first step

Questions?